Adding CycleStart,Stop,Pause and EStop External Buttons

Use this example at your own risk and practice all machine safety procedures. Example of adding physical external buttons. Review the example program “ExternalButtons.c” that’s in Kmotion main directory folder in the C Programs folder.

1) Open your main INIT c program using KMotion.

2) Define button bits by adding the following at top of INIT file right below #include "KMotionDef.h"

```
#define CYCLESTARTBIT 20
#define FEEDHOLDBIT 21
#define HALTBIT 22
#define ESTOP 23
```

3) Add the following lines below your define button bits.(Note: Some of this may already be in your INIT C program.)

```
// function prototypes for compiler
int DoPC(int cmd);
int DoPCFloat(int cmd, float f);
int Debounce(int n, int *cnt, int *last, int *lastsolid);

// state variables for switch debouncing
int flast=0,flastsolid=-1,fcount=0;
int clast=0,clastsolid=-1,ccount=0;
int elast=0,elastsolid=-1,ecount=0;
int hlast=0,hlastsolid=-1,hcount=0;
int rlast=0,rlastsolid=-1,rcount=0;
int zlast=0,zlastsolid=-1,zcount=0;
```

4) Add the following below motor configs towrd bottom of program. (Note: The loop line of code “for (;;) // loop forever” may not be needed.)

```
}
            for (;;) // loop forever
                  WaitNextTimeSlice();

            // Handle FeedHold/Resume
            result = Debounce(ReadBit(FEEDHOLDBIT),&fcount,&flast,&flastsolid);
            if  (result == 1)
            {
                  if (CS0_StoppingState == 0)
                        StopCoordinatedMotion();
                  else
                        ResumeCoordinatedMotion();
            }

                  // Handle Cycle Start
            result = Debounce(ReadBit(CYCLESTARTBIT),&ccount,&clast,&clastsolid);
            if  (result == 0)
            {
                  DoPC(PC_COMM_EXECUTE);
            }

            // Handle ESTOP
            result = Debounce(ReadBit(ESTOP),&ecount,&elast,&elastsolid);
            if  (result == 1)
            {
                  DoPC(PC_COMM_ESTOP);
                  SetStateBit(44,0);
            }
            // Handle HALT
            result = Debounce(ReadBit(HALTBIT),&hcount,&hlast,&hlastsolid);
            if  (result == 1)
            {
                  DoPC(PC_COMM_HALT);
                  SetStateBit(44,0);
            }
      }
}
```

5) If your C program does not have the following code, add it at the bottom.

```
// Put a Float as a parameter and pass the command to the App
            int DoPCFloat(int cmd, float f)
            {
                  int result;
                  persist.UserData[PC_COMM_PERSIST+1] = *(int*)&f;
                  return DoPC(cmd);
            }

            // Pass a command to the PC and wait for it to handshake
            // that it was received by either clearing the command
            // or changing it to a negative error code
            int DoPC(int cmd)
            {
                  int result;
                  persist.UserData[PC_COMM_PERSIST]=cmd;

            do
            {
                  WaitNextTimeSlice();
            }while (result=persist.UserData[PC_COMM_PERSIST]>0);

            printf("Result = %d\n",result);

            return result;
      }

// Debounce a bit
// return 1 one time when first debounced high
// return 0 one time when first debounced low
// return -1 otherwise
#define DBTIME 300

int Debounce(int n, int *cnt, int *last, int *lastsolid)
{
int v = -1;

      if (n == *last)  // same as last time?
      {
            if (*cnt == DBTIME-1)
            {
                  if (n != *lastsolid)
                  {
                        v = *lastsolid = n;  // return debounced value
                  }
            }
            if (*cnt < DBTIME)      (*cnt)++;
      }
      else
      {
            *cnt = 0;  // reset count
      }
      *last = n;
      return v;
}
```

6) Compile to make sure there is no errors.
7) Save and test in KMotion CNC.